Table 50. Vector table (Cat.3 devices) (continued)

| Position | Priority | Type of priority | Acronym | Description | Address |
|---|---|---|---|---|---|
| 52 | 62 | settable | AES | AES global interrupt | 0x0000_011C |
| 53 | 63 | settable | COMP_ACQ | Comparator Channel Acquisition Interrupt | 0x0000_0120 |

Table 51. Vector table (Cat.4, Cat.5 and Cat.6 devices)

| Position | Priority | Type of priority | Acronym | Description | Address |
|---|---|---|---|---|---|
| - | - | - | - | Reserved | 0x0000_0000 |
| - | -3 | fixed | Reset | Reset | 0x0000_0004 |
| - | -2 | fixed | NMI_Handler | Non maskable interrupt. The RCC Clock Security System (CSS) is linked to the NMI vector. | 0x0000_0008 |
| - | -1 | fixed | HardFault_Handler | All class of fault | 0x0000_000C |
| - | 0 | settable | MemManage_Handler | Memory management | 0x0000_0010 |
| - | 1 | settable | BusFault_Handler | Pre-fetch fault, memory access fault | 0x0000_0014 |
| - | 2 | settable | UsageFault_Handler | Undefined instruction or illegal state | 0x0000_0018 |
| - | - | - | - | Reserved | 0x0000_001C - 0x0000_002B |
| - | 3 | settable | SVC_Handler | System service call via SWI instruction | 0x0000_002C |
| - | 4 | settable | DebugMon_Handler | Debug Monitor | 0x0000_0030 |
| - | - | - | - | Reserved | 0x0000_0034 |
| - | 5 | settable | PendSV_Handler | Pendable request for system service | 0x0000_0038 |
| - | 6 | settable | SysTick_Handler | System tick timer | 0x0000_003C |
| 0 | 7 | settable | WWDG | Window Watchdog interrupt | 0x0000_0040 |
| 1 | 8 | settable | PVD | PVD through EXTI Line16 detection interrupt | 0x0000_0044 |
| 2 | 9 | settable | TAMPER_STAMP | Tamper, LSECSS and TimeStamp through EXTI line19 interrupts | 0x0000_0048 |
| 3 | 10 | settable | RTC_WKUP | RTC Wakeup through EXTI line20 interrupt | 0x0000_004C |
| 4 | 11 | settable | FLASH | Flash global interrupt | 0x0000_0050 |
| 5 | 12 | settable | RCC | RCC global interrupt | 0x0000_0054 |
| 6 | 13 | settable | EXTI0 | EXTI Line0 interrupt | 0x0000_0058 |
| 7 | 14 | settable | EXTI1 | EXTI Line1 interrupt | 0x0000_005C |
| 8 | 15 | settable | EXTI2 | EXTI Line2 interrupt | 0x0000_0060 |
| 9 | 16 | settable | EXTI3 | EXTI Line3 interrupt | 0x0000_0064 |

Table 51. Vector table (Cat.4, Cat.5 and Cat.6 devices) (continued)

| Position | Priority | Type of priority | Acronym | Description | Address |
|---|---|---|---|---|---|
| 10 | 17 | settable | EXTI4 | EXTI Line4 interrupt | 0x0000_0068 |
| 11 | 18 | settable | DMA1_Channel1 | DMA1 Channel1 global interrupt | 0x0000_006C |
| 12 | 19 | settable | DMA1_Channel2 | DMA1 Channel2 global interrupt | 0x0000_0070 |
| 13 | 20 | settable | DMA1_Channel3 | DMA1 Channel3 global interrupt | 0x0000_0074 |
| 14 | 21 | settable | DMA1_Channel4 | DMA1 Channel4 global interrupt | 0x0000_0078 |
| 15 | 22 | settable | DMA1_Channel5 | DMA1 Channel5 global interrupt | 0x0000_007C |
| 16 | 23 | settable | DMA1_Channel6 | DMA1 Channel6 global interrupt | 0x0000_0080 |
| 17 | 24 | settable | DMA1_Channel7 | DMA1 Channel7 global interrupt | 0x0000_0084 |
| 18 | 25 | settable | ADC1 | ADC1 global interrupt | 0x0000_0088 |
| 19 | 26 | settable | USB HP | USB High priority interrupt | 0x0000_008C |
| 20 | 27 | settable | USB_LP | USB Low priority interrupt | 0x0000_0090 |
| 21 | 28 | settable | DAC | DAC interrupt | 0x0000_0094 |
| 22 | 29 | settable | COMP/CA | Comparator wakeup through EXTI line (21 and 22) interrupt/Channel acquisition interrupt | 0x0000_0098 |
| 23 | 30 | settable | EXTI9_5 | EXTI Line[9:5] interrupts | 0x0000_009C |
| 24 | 31 | settable | LCD | LCD global interrupt | 0x0000_00A0 |
| 25 | 32 | settable | TIM9 | TIM10 global interrupt | 0x0000_00A4 |
| 26 | 33 | settable | TIM10 | TIM10 global interrupt | 0x0000_00A8 |
| 27 | 34 | settable | TIM11 | TIM11 global interrupt | 0x0000_00AC |
| 28 | 35 | settable | TIM2 | TIM2 global interrupt | 0x0000_00B0 |
| 29 | 36 | settable | TIM3 | TIM3 global interrupt | 0x0000_00B4 |
| 30 | 37 | settable | TIM4 | TIM4 global interrupt | 0x0000_00B8 |
| 31 | 38 | settable | I2C1_EV | $I^2C1$ event interrupt | 0x0000_00BC |
| 32 | 39 | settable | I2C1_ER | $I^2C1$ error interrupt | 0x0000_00C0 |
| 33 | 40 | settable | I2C2_EV | $I^2C2$ event interrupt | 0x0000_00C4 |
| 34 | 41 | settable | I2C2_ER | $I^2C2$ error interrupt | 0x0000_00C8 |
| 35 | 42 | settable | SPI1 | SPI1 global interrupt | 0x0000_00CC |
| 36 | 43 | settable | SPI2 | SPI2 global interrupt | 0x0000_00D0 |
| 37 | 44 | settable | USART1 | USART1 global interrupt | 0x0000_00D4 |
| 38 | 45 | settable | USART2 | USART2 global interrupt | 0x0000_00D8 |
| 39 | 46 | settable | USART3 | USART3 global interrupt | 0x0000_00DC |
| 40 | 47 | settable | EXTI15_10 | EXTI Line[15:10] interrupts | 0x0000_00E0 |
| 41 | 48 | settable | RTC_Alarm | RTC Alarms (A and B) through EXTI line17 interrupt | 0x0000_00E4 |

Table 51. Vector table (Cat.4, Cat.5 and Cat.6 devices) (continued)

| Position | Priority | Type of priority | Acronym | Description | Address |
|---|---|---|---|---|---|
| 42 | 49 | settable | USB_FS_WKUP | USB Device FS Wakeup through EXTI line18 interrupt | 0x0000_00E8 |
| 43 | 50 | settable | TIM6 | TIM6 global interrupt | 0x0000_00EC |
| 44 | 51 | settable | TIM7 | TIM7 global interrupt | 0x0000_00F0 |
| 45 | 52 | settable | SDIO | SDIO Global interrupt | 0x0000_00F4 |
| 46 | 53 | settable | TIM5 | TIM5 Global interrupt | 0x0000_00F8 |
| 47 | 54 | settable | SPI3 | SPI3 Global interrupt | 0x0000_00FC |
| 48 | 55 | settable | UART4 | UART4 Global interrupt | 0x0000_0100 |
| 49 | 56 | settable | UART5 | UART5 Global interrupt | 0x0000_0104 |
| 50 | 57 | settable | DMA2_CH1 | DMA2 Channel 1 interrupt | 0x0000_0108 |
| 51 | 58 | settable | DMA2_CH2 | DMA2 Channel 2 interrupt | 0x0000_010C |
| 52 | 59 | settable | DMA2_CH3 | DMA2 Channel 3 interrupt | 0x0000_0110 |
| 53 | 60 | settable | DMA2_CH4 | DMA2 Channel 4 interrupt | 0x0000_0114 |
| 54 | 61 | settable | DMA2_CH5 | DMA2 Channel 5 interrupt | 0x0000_0118 |
| 55 | 62 | settable | AES | AES global interrupt | 0x0000_011C |
| 56 | 63 | settable | COMP_ACQ | Comparator Channel Acquisition Interrupt | 0x0000_0120 |

## 10.2 External interrupt/event controller (EXTI)

The external interrupt/event controller consists of up to 24 (or 23 for Cat.1 and Cat.2 devices) edge detectors for generating event/interrupt requests. Each input line can be independently configured to select the type (event or interrupt) and the corresponding trigger event (rising edge, falling edge or both). Each line can also be masked independently. A pending register maintains the status line of the interrupt requests.

### 10.2.1 Main features

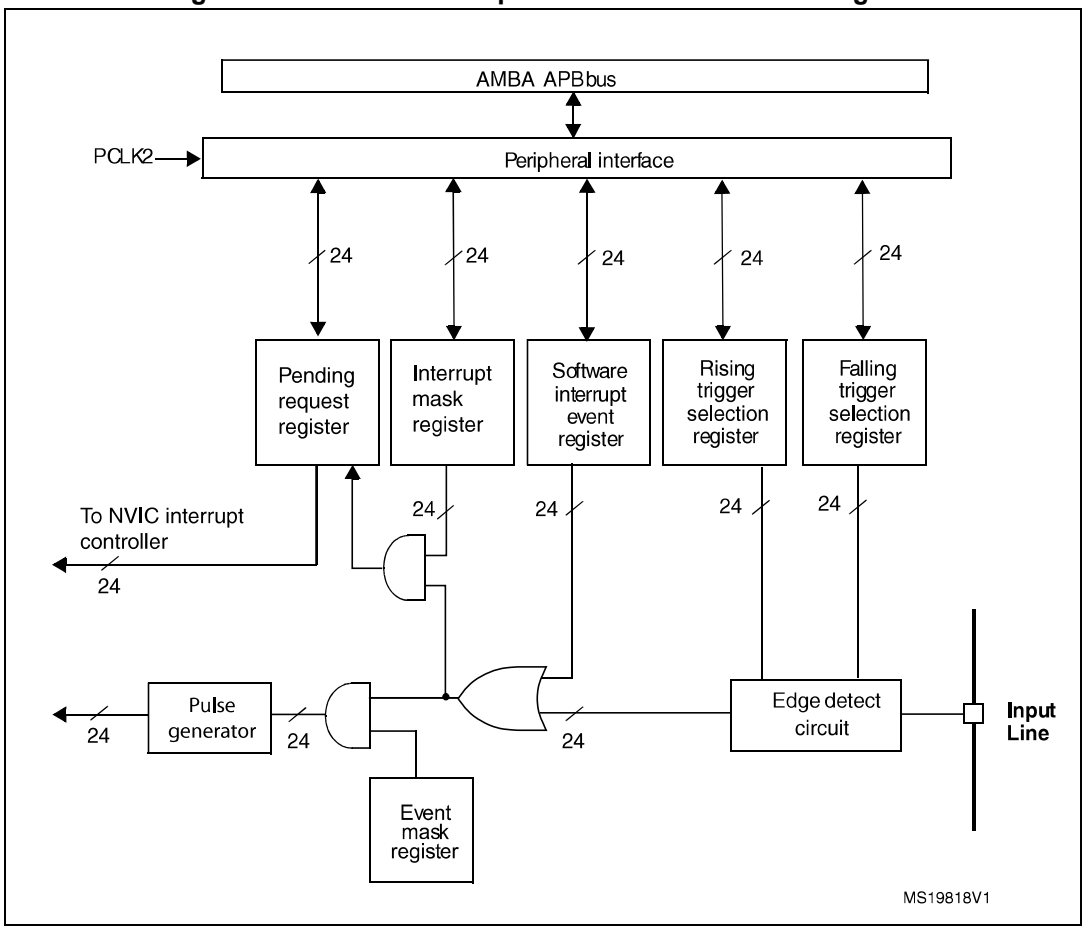
The main features of the EXTI controller are the following:

- Independent trigger and mask on each interrupt/event line
- Dedicated status bit for each interrupt line
- Generation of up to 24 (or 23 for Cat.1 and Cat.2 devices) software event/interrupt requests
- Detection of external signals with a pulse width lower than the APB2 clock period. Refer to the electrical characteristics section of the STM32L1xxxx datasheet for details on this parameter.

### 10.2.2 Block diagram

The block diagram is shown in *Figure 32*.

**Figure 32. External interrupt/event controller block diagram**




## 10.2.3 Wakeup event management

The STM32L1xxxx is able to handle external or internal events in order to wake up the core (WFE). The wakeup event can be generated by either:

- enabling an interrupt in the peripheral control register but not in the NVIC, and enabling the SEVONPEND bit in the Cortex®-M3 system control register. When the MCU resumes from WFE, the peripheral interrupt pending bit and the peripheral NVIC IRQ channel pending bit (in the NVIC interrupt clear pending register) have to be cleared.
- or configuring an external or internal EXTI line in event mode. When the CPU resumes from WFE, it is not necessary to clear the peripheral interrupt pending bit or the NVIC IRQ channel pending bit as the pending bit corresponding to the event line is not set.

To use an external line as a wakeup event, refer to *Section 10.2.4: Functional description*.

## 10.2.4 Functional description

To generate the interrupt, the interrupt line should be configured and enabled. This is done by programming the two trigger registers with the desired edge detection and by enabling the interrupt request by writing a ‘1 to the corresponding bit in the interrupt mask register. When the selected edge occurs on the external interrupt line, an interrupt request is generated. The pending bit corresponding to the interrupt line is also set. This request is reset by writing a ‘1 into the pending register.